

# A New Systems Engineering Model and an Old, Familiar Friend

Mary Redshaw

**T**he arrival of the 21st century and the aftermath of 9/11 brought with them many fundamental changes that impact defense acquisition programs. Our systems are becoming increasingly complex and costly. Developing and sustaining complex systems while achieving cost-wise readiness demands excellence in systems engineering. Because we require a disciplined development process as never before, a great deal of attention and energy is focused on efforts to revitalize systems engineering within the Department of Defense. Does “revitalization” mean that we have to learn a whole new way of doing business?

The 16 systems engineering processes described in the *Defense Acquisition Guidebook (DAG)* do not—at first glance—appear “familiar” to those who learned the legacy systems engineering process model taught by the Defense Acquisition University in recent years. This article models those 16 processes in a way that presents our familiar friend—the legacy model—in a powerful new construct.

## Revitalizing Systems Engineering

The under secretary of defense (acquisition, technology and logistics) issued a policy memorandum in February 2004 that stressed the importance of systems engineering in defense acquisition programs and the need to “drive good systems engineering practices back into the way we do business.” That statement highlights the fact that the DoD is revitalizing its internal practices in a discipline in which it has excelled in the past.

The term “systems engineering” was first coined at Bell Telephone Laboratories in the early 1940s, and DoD began practicing the concept later that decade with the initial development of missiles and missile-defense systems. Systems engineering started gaining momentum following World War II. Because of its role in acquiring and developing large-scale, complex systems, DoD led the way in codifying the fledgling discipline by developing and releasing the first systems engineering standard in 1969. The principles in that baseline military standard (and later revisions) are still valid. Efforts aimed at revitalizing sys-

**The legacy model is elegant in its simplicity—simplicity that makes it easy to remember while conveying some of the complexities of the systems engineering problem-solving methodology.**

tems engineering should retain those aspects of the discipline that have proved successful in developing complex systems in the past—perhaps in a framework that has evolved over time—and avoid throwing out the baby with the bath water.

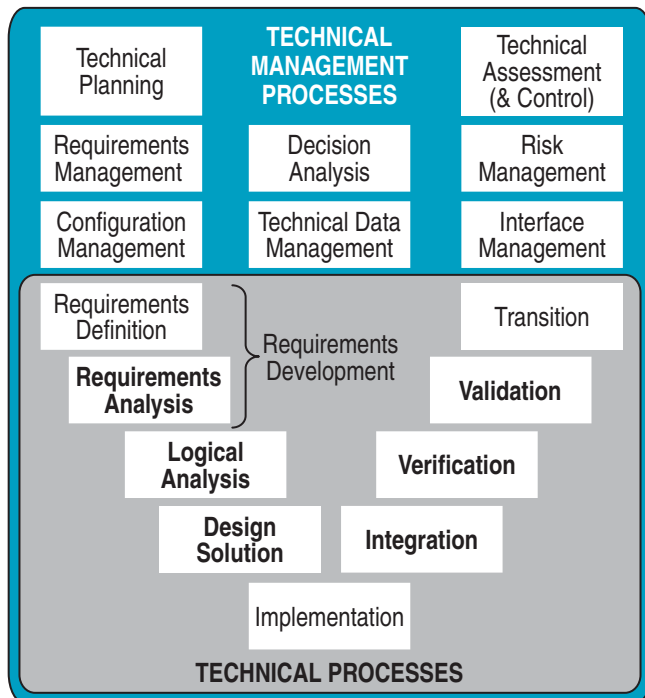
## The Demise—and Resurrection—of a Standard

In 1969 the U.S. Air Force developed the baseline military standard, Systems Engineering Management, MIL STD 499 (USAF). The standard was approved by DoD and was considered for possible conversion to a fully coordinated document mandatory for use by all DoD agencies. Revision A was published in 1974, again primarily for use by the Air Force. Later acquisition reform in the early 1990s emphasized use of commercial standards when available and appropriate—as the precursor of performance-based acquisition initiatives.

In 1994, then-Defense Secretary William Perry issued a policy memorandum barring the use of military specifications and standards on DoD acquisition programs unless a waiver was granted by the milestone decision au-

*Redshaw is a professor at DAU, Mid-Atlantic Region. She holds a doctorate in engineering management and is a certified systems engineering professional.*

**FIGURE 1. Comprehensive Systems Engineering Process Model**



thority. Revision B of the DoD systems engineering standard was intended for use on all DoD programs and was circulated as a coordination draft to a wide audience (including industry partners) in 1992. Because of the ongoing reform initiatives leading to the Perry Memorandum, MIL STD 499B was never approved for DoD release, and MIL STD 499A (USAF) was subsequently cancelled without replacement in 1995.

Because no commercial systems engineering standards existed in 1994, the coordination draft of MIL STD 499B was embraced by U.S. industry standards bodies as the basis for two standards (IEEE-1220 and EIA-632), both of which represented fairly minor modifications of the military standard. Since that time, DoD acquisition organizations have used industry standards as the framework for developing their own systems engineering guides and handbooks. The practice of systems engineering within DoD became increasingly fragmented by proliferating standards, models, and process improvement frameworks.

However, the pendulum is starting to swing in the opposite direction—in both industry and government circles. Over the past few years, commercial systems and software engineering standards have begun slowly converging toward a single harmonized international standard. More recently, the U.S. Air Force released a new draft (Revision C) of the military standard, intended to support acquisitions by the Space Missile Systems Center within the Air Force Space Command. The coordination draft of MIL STD 499C (USAF) was released on March 24, 2005, with

the intent that it be made available for use by all departments and agencies of the DoD. Invoking it as a compliance standard on DoD contracts became a possibility five days later with issuance of a USD(AT&L) policy memorandum on March 29, 2005 that allows program managers the flexibility to require conformance to military standards and specifications where appropriate—without having to seek a waiver from the milestone decision authority.

### **The Legacy DoD Systems Engineering Model**

Although it was never approved for DoD use, current members of the acquisition community have been exposed to concepts and artifacts from the coordination draft of MIL STD 499B, including the legacy systems engineering process model used by DAU in its courses today. I was first exposed to that model when I attended the 20-week program management course at the Defense Systems Management College in 1992; and I have been teaching the same model for a little over two years as an instructor at DAU. To me, the legacy model is an old, familiar friend. Ironically, this old friend appears in the newly released coordination draft of MIL STD 499C.

The legacy model is elegant in its simplicity—simplicity that makes it easy to remember while conveying some of the complexities of the systems engineering problem-solving methodology—such as its iterative and recursive nature. It contains three primary sequential process steps: requirements analysis, functional analysis and allocation, and synthesis. The model also depicts a block, interfacing with all three process steps, entitled “Systems Analysis and Control,” which is a compilation of management activities and tools (e.g., for decision analysis, assessment, and control). At a high level, the model captures the sequential order of the primary steps, their interface with the management activities throughout their application, and recursive loops between process pairs that ensure all requirements are completely defined, traced, and verified. One major disadvantage of the model is that the verification loop does not adequately convey the role of test planning, testing, and evaluation of results as integral parts of the development process.

Perhaps to overcome the deficiency noted above, variations of V-shaped models have lately become prevalent within industry systems-engineering frameworks, including the first international consensus systems engineering standard (ISO/IEC-15288, released in 2002). The new *DAG* describes 16 “generic systems engineering processes.” In lieu of the legacy model, the *DAG* portrays a series of five sets of phase-based activities arranged in V-shaped patterns, as does the Integrated Defense Acquisition, Technology, and Logistics Life Cycle Management Framework (the Wall Chart). Unfortunately, neither the *DAG* nor the Wall Chart provides a single generic model for instructional purposes. As part of my work in devel-

## *You're the Judge: The Verdict* (from page 38)

Yes, Ms. Bryan has a problem, and she did commit a crime.

On April 7, 2004, Bryan pleaded guilty to one count of violating 18 U.S.C. § 207(a)(1), one of the post-government service employment communication restrictions. The Eastern District of Virginia handled the prosecution.

This law prohibits former federal personnel from representing someone else before the federal government on particular matters involving specific parties that he or she worked on personally and substantially while in the federal government with the intent to influence the government's decision.

In her official capacity, Bryan worked on the contract between the government and Software Professionals and its terms, including termination. She didn't commit a crime when she went to work for Software Professionals. Only when she represented Software Professionals before the government to extend the term of the existing contract did she violate 18 U.S.C. 207(a)(1) because it was a matter she had originally negotiated as a federal employee.

On July 23, 2004, Bryan was sentenced to two years' supervised probation, substance abuse treatment, and a special assessment.

took some literary license with respect to a couple of the DAG processes, as shown in Figure 1 and described below:

- The parenthetical “& Control” is added to the technical assessment process, indicating the need for corrective action if assessment of project status or outcomes indicates deviation from planning baselines.
- The requirements development process is decomposed into two subordinate processes to capture the overlap of the acquisition/systems-engineering domain with the Joint Capabilities Integration and Development System (JCIDS).
- The technical management processes in the CSEP model are equivalent to the systems analysis and control portion of the legacy model. Note that in the CSEP model the technical processes are always implemented within the encompassing framework of the technical management processes. Collectively the technical management processes form the executive—or control—logic that steers system development to meet project or phase objectives.

The technical processes are depicted in a V-shaped pattern. Again for ease of reference—and as a description of its function and power—I call this V-shaped model of the technical processes the V-9 Engine (Figure 2). The blue blocks in the V-9 Engine capture the legacy model's three primary sequential process steps on the left-hand side, plus associated steps inferred or adapted from the legacy model and the ISO/IEC 15288 model, respectively, on the right-hand side.

### **Powerful Visualization with the V-9 Engine**

The V-9 Engine provides a powerful visualization of key process interfaces. The concept of interfaces between different levels in the system hierarchy is particularly important in the system-of-systems or net-centric context. It is important that the systems engineer responsible for developing a system or subordinate element view it from the outside, or from the perspective of the larger architecture in which it is intended to operate.

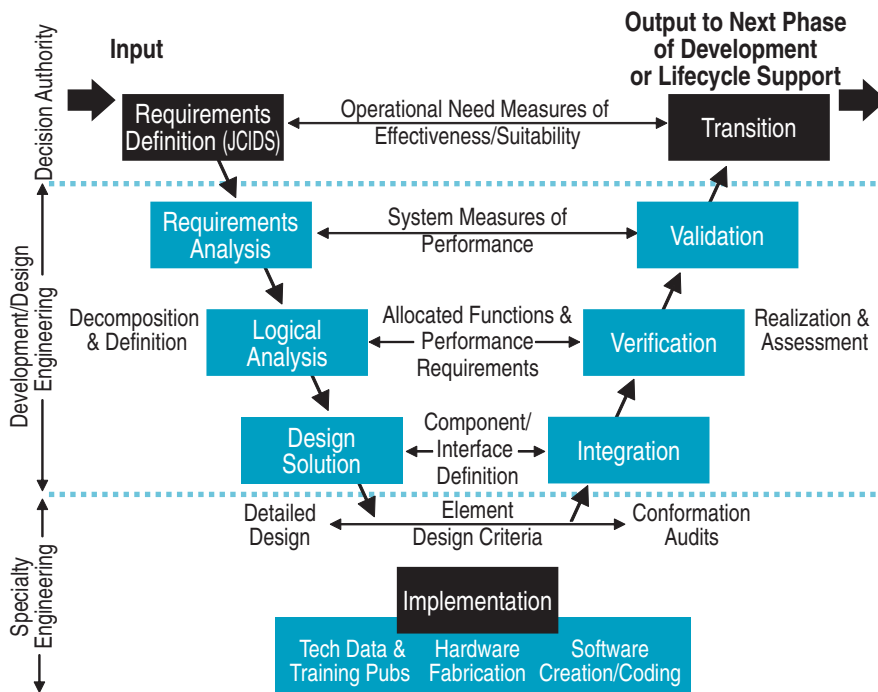
The V-9 Engine illustrates domains of responsibility within the technical processes. The subdivision of the requirements development process into two subordinates portrays interfaces of a project team with the JCIDS process, with project or engineering managers at a higher level in the system hierarchy, or with the acquiring organization where an acquirer-supplier agreement exists. The results of the first subordinate process—requirements definition—governs the development (or manufacturing) effort and establishes the “handshake” regarding project scope and deliverables between the project decision authority and the development team. At the end of a phase of development, review of products and test results during the transition process allows the decision authority to determine if all requirements and agreements have been met;

oping a new systems engineering course consistent with the direction taken by DoD—and largely to help myself understand and explain that new direction—I developed a new model. This model captures the 16 processes listed in the DAG, provides a generic representation of the series of phase-based activities and can be correlated to the legacy DoD systems engineering model. For ease of reference in discussion, I call it the Comprehensive Systems Engineering Process (CSEP) model.

### **Proposed: A New Model for DoD Systems Engineering**

In Chapter 4 (“Systems Engineering”), the DAG introduces eight technical management processes and eight technical processes. In modeling those 16 processes—and in developing a generic representation of the phase-based series of V-shaped activities—I adapted a model contained in ISO/IEC 15288. To reconcile with the legacy model, I

**FIGURE 2. V-9 Process Interactions**



if further system development is warranted; and if the system is ready to proceed to the next work effort, phase, or acquisition life-cycle function (e.g., production, deployment, operation).

Another key process interface occurs at the point where the design is implemented. This is the level at which the systems engineer and component design specialist(s) identify and resolve technical issues and select workable solutions that will not jeopardize the overall system design, capabilities, performance, or suitability. The level of the system hierarchy at which the design is handed over to specialists for implementation is project-dependent. However, in all cases, the systems engineer monitors the implementation of system elements as they affect overall design, performance, cost, and schedule.

**While retaining the same process steps and attributes of the legacy model, the CSEP model suggests additional valuable information.**

Finally, the V-9 Engine highlights some of the important characteristics of the technical processes, including the sequential order of process application (or completion). In the top-down application on the left-hand side, measurable criteria are documented at each level of system decomposition and definition—forming the basis for assessment during bottom-up system realization on the right-hand side. Requirements are traced throughout the iterative and recursive application of this problem-solving “engine” to ensure complete and balanced coverage of input and derived requirements to the system and lower elements in the system hierarchy. In the CSEP model, the V-9 is the “engine in systems engineering.”

### Comparing Legacy and CSEP

Comparing the legacy model to the CSEP model—and its constituent V-9

Engine—is analogous to comparing a view of a piece of electronics equipment with the face plate installed versus removed. Viewing the equipment with the face plate removed reveals the connections and interfaces inside. Seeing those connections increases the understanding of how the electronics equipment functions—or in this case—how the overall process is intended to work.

While retaining the same process steps and attributes of the legacy model, the CSEP model suggests additional valuable information regarding the encompassing and executive nature of the technical management processes, relationships among the technical processes, domains of responsibility, the importance of test planning during system definition, and the integration of test and evaluation activities as part of system development. Understanding the correlation of the legacy model to the CSEP model is valuable also. A practitioner familiar with the legacy model—or someone seeing it for the first time in the coordination draft of MIL-STD-499C—can readily understand the models presented in this article. Since the proposed CSEP model was adapted from one in an international consensus standard, a practitioner using either the source or derived model will quickly also recognize the correlation of the processes in the other.

The author welcomes questions and comments. She can be contacted at [mary.redshaw@dau.mil](mailto:mary.redshaw@dau.mil).